



GEMS: A Fully Integrated PETSc-Based Solver for Coupled Cardiac Electromechanics and Bidomain Simulations

Sander Arens¹, Hans Dierckx¹ and Alexander V. Panfilov^{1,2*}

¹ Department of Physics and Astronomy, Ghent University, Ghent, Belgium, ² Laboratory of Computational Biology and Medicine, Ural Federal University, Ekaterinburg, Russia

OPEN ACCESS

Edited by:

Mariano Vázquez,
Barcelona Supercomputing Center,
Spain

Reviewed by:

Jazmin Aguado-Sierra,
Barcelona Supercomputing Center,
Spain

Constantine Butakoff,
Universidad Pompeu Fabra, Spain

*Correspondence:

Alexander V. Panfilov
Alexander.Panfilov@ugent.be

Specialty section:

This article was submitted to
Biophysics,
a section of the journal
Frontiers in Physiology

Received: 14 January 2018

Accepted: 20 September 2018

Published: 16 October 2018

Citation:

Arens S, Dierckx H and Panfilov AV
(2018) GEMS: A Fully Integrated
PETSc-Based Solver for Coupled
Cardiac Electromechanics and
Bidomain Simulations.
Front. Physiol. 9:1431.
doi: 10.3389/fphys.2018.01431

Cardiac contraction is coordinated by a wave of electrical excitation which propagates through the heart. Combined modeling of electrical and mechanical function of the heart provides the most comprehensive description of cardiac function and is one of the latest trends in cardiac research. The effective numerical modeling of cardiac electromechanics remains a challenge, due to the stiffness of the electrical equations and the global coupling in the mechanical problem. Here we present a short review of the inherent assumptions made when deriving the electromechanical equations, including a general representation for deformation-dependent conduction tensors obeying orthotropic symmetry, and then present an implicit-explicit time-stepping approach that is tailored to solving the cardiac mono- or bidomain equations coupled to electromechanics of the cardiac wall. Our approach allows to find numerical solutions of the electromechanics equations using stable and higher order time integration. Our methods are implemented in a monolithic finite element code GEMS (Ghent Electromechanics Solver) using the PETSc library that is inherently parallelized for use on high-performance computing infrastructure. We tested GEMS on standard benchmark computations and discuss further development of our software.

Keywords: cardiac arrhythmias, electromechanics, cardiac modeling, ionic models, anatomical models

1. INTRODUCTION

The heart is an electromechanical pump whose mechanical contraction is initiated by electrical activation, in a process called excitation-contraction coupling. In normal circumstances, contraction is highly synchronized, resulting in an efficient throughput of oxygenated blood to the body. Failure in doing so can lead to sudden cardiac death. The contraction also affects excitation via the process called mechano-electrical feedback. An example of mechano-electrical feedback that has fatal consequences is *commotio cordis* (Maron and Estes, 2010), a long-known (Akenside, 1763; Meola, 1879; Nesbitt et al., 2001) phenomenon where a blow to the chest (even without damaging the heart) may cause ventricular fibrillation. *Commotio cordis* is still an important cause of sudden cardiac death in young athletes (Maron, 2003). The underlying mechanism of mechano-electrical feedback is caused by several factors, including stretch-activated ionic channels (Kohl et al., 2001). Although much is already known about the subcellular contributions to mechano-electrical feedback (Quinn et al., 2014), it is still unclear how these translate to macroscopic scales. Computational models can further help understand the mechanisms and consequences of cardiac mechano-electrical feedback up to the organ level.

The heart is mostly modeled as a continuum via partial differential equations (PDEs). For the spatial coupling between cells, the cardiac mono- or bidomain equations (Keener and Sneyd, 2009) are commonly used, in which any specific model for individual cardiac cells can be inserted. For the mechanical problem, the most commonly used are the PDEs of finite (hyper)elasticity (Nash and Hunter, 2000). The joint solution of these equations is a considerable numerical challenge. The difficulties largely originate from the different physical interactions that occur on a wide range of spatial and temporal scales (Plank et al., 2008; Keyes et al., 2013). The multiphysics nature makes it impossible to use a general-purpose black-box solver for this task. Solvers can only be optimal if they use as much information as possible about the problem. For example, implicit/explicit integrators need to know which processes are fast or slow, field-split preconditioners (Brown et al., 2012; Liu and Keyes, 2015) need to be able to extract fields belonging to different physics, and multigrid (Briggs et al., 2000; Trottenberg et al., 2000) and domain decomposition (Quarteroni and Valli, 1999; Smith et al., 2004) solvers need information about the meshes and discretizations.

In recent years, computational modeling of cardiac electromechanics has become an active field of research see e.g., (Göktepe and Kuhl, 2010; Lafortune et al., 2012; Land et al., 2012; Fritz et al., 2014; Rossi et al., 2014; Franzone et al., 2015; Augustin et al., 2016). However, different groups often use different descriptions for the same problems with different forms for deformation-dependent conduction tensors and sometimes convective terms in the undeformed configuration. In addition, current electromechanics codes are often the result of *ad hoc* coupling methods between the electrophysiology and finite elasticity codes, limiting time integration to only first order numerical schemes and poor stability, although some approaches are known to address these stability issues (Niederer and Smith, 2008; Pathmanathan and Whiteley, 2009). This problem is common in other fields that use multiphysics (Keyes et al., 2013).

Our contributions in this paper are the following. First, we give a consistent derivation of the continuum equations of coupled electromechanics of the heart based on basic principles from geometry and physics and the clarification of the constitutive equations used. From this we show that there are no convective terms in the undeformed configuration and that the variety of deformation-dependent conduction tensors from literature are all special cases of a more general form that we present here. Second, we generalize Euler-based implicit-explicit schemes for electromechanics to higher order implicit-explicit Runge-Kutta schemes, based on the knowledge of fast/slow dynamics. Third, we explain on how to solve the resulting non-linear implicit equations from a general multiphysics perspective.

This paper is structured as follows. In section 2 we introduce the necessary notations and concepts and present the strong and weak form for the continuum electromechanics equations, followed by a brief discussion on how to discretize the weak form equations using finite elements in section 3. Next, we discuss on how to discretize the electromechanics equations in time using implicit-explicit schemes and how to solve the resulting non-linear equations in 4. Finally, in section 5 we explain how we

implemented this using PETSc (Balay et al., 1997, 2016a,b) in our GEMS (Ghent ElectroMechanics Solver) code, and give examples of numerical results in section 6.

2. PHYSICS

In this section we introduce the mathematical basis for physical modeling in the moving domain, distinguishing between the Eulerian and Lagrangian viewpoints. Then we show how the balance equations (i.e., physical conservation laws) need to be closed by constitutive equations. By imposing symmetry (e.g., a locally uniaxial medium), the constitutive equations involving tensors cannot be chosen freely, but need to be of certain form which we here propose and discuss. We conclude by splitting the equations in fast and slow components, which will be respectively treated implicitly and explicitly during time stepping in section 4. At the end of this section, we will have cast the modeling equations in variational form, suitable for use in the finite element approach.

2.1. Definitions and Notation for Geometrical Concepts

To formulate the problem of electromechanics, it is important to understand the underlying geometry. Since we will consider continuum equations here, it is natural to consider them on a manifold, i.e., a “curved” space which locally resembles Euclidean space. For additional background material we refer to Marsden and Hughes (1994) and Frankel (2012).

Let \mathcal{B} be the *material manifold* of dimension m . This is a reference manifold for our body. For an excitable surface, $m = 2$ and for a three-dimensional tissue, $m = 3$. On every patch of \mathcal{B} , we define material coordinates X^I , $I = 1, \dots, m$.

The space in which the body moves is given by the *spatial manifold* \mathcal{S} (which is sometimes called the ambient or target manifold), of dimension n . For example, if an excitable surface is restricted to move in a plane, $n = 2$. However, in the general case where the tissue can move in 3D, $n = 3$. On every patch of \mathcal{S} , we define spatial coordinates x^i .

We will assume that we have a metric for these manifolds, which we denote by resp. G and g , so that we have Riemannian manifolds. In the simplest case (which we will use further) \mathcal{S} will be n -dimensional Euclidean space, such that x^i are Cartesian coordinates x, y, z , and \mathcal{B} will be an open subset of Euclidean m -dimensional space. However, a non-Euclidean metric on \mathcal{B} can be important in growth and remodeling phenomena (Ozakin and Yavari, 2010), e.g., hypertrophy and thermoelasticity (Yavari, 2010).

A *configuration* of \mathcal{B} is a mapping $\phi : \mathcal{B} \rightarrow \mathcal{S}$ which represents the deformation of the body and we will often use the notation $x^i = \phi^i$. The set of all configurations of \mathcal{B} is called the *configuration space* \mathcal{C} and is an infinite-dimensional manifold.

The tangent map $T\phi : T\mathcal{B} \rightarrow T\mathcal{S}$, $T\phi(X, V) = (\phi(X), D\phi(V))$ is called the *deformation gradient* F and is $F_I^i = \frac{\partial \phi^i}{\partial X^I}$ in components. This tells us how a tangent vector at a point $X \in \mathcal{B}$ transforms under ϕ .

Another important concept is the *deformation tensor* C , which is the pullback of the metric g : $C = \phi^*g$, or in components $C_{IJ} = F_I^i g_{ij} F_J^j$. Note that the squared infinitesimal distance between nearby points with coordinates X^I and $X^I + dX^I$ or x^i and $x^i + dx^i$ is $ds^2 = g_{ij} dx^i dx^j = C_{IJ} dX^I dX^J$, showing that C_{IJ} is a measure for how length and angles between fixed pairs of points in the tissue change under a deformation. If we pull back the volume form dv on \mathcal{S} to \mathcal{B} , we get $\phi^*dv = JdV$, where dV is the volume form on \mathcal{B} and $J = \frac{\sqrt{\det g}}{\sqrt{\det G}} \det F$ the Jacobian of the deformation.

The strain in the tissue will depend on how the current length and angles relate to the reference case, which is quantified by the *strain tensor* $E = \frac{\phi^*g - G}{2}$. Since ϕ is an isometry only if $\phi^*g = G$, E measures the deviation between the current deformation and an isometry.

In cardiac contraction, the configuration (or deformation) ϕ is time-dependent, which can be represented by a curve \mathcal{C} in configuration space, i.e., a mapping $\mathbb{R} \rightarrow \mathcal{C}$; $t \rightarrow \phi_t$, called the *motion*. The *material velocity* and *acceleration* are then defined to be respectively the first and second time derivatives of the motion. Their components are given by $V^i = \frac{\partial \phi^i}{\partial t}$ and $A^i = \frac{\partial V^i}{\partial t} + (\gamma_{jk}^i \circ \phi) V^j V^k$, where γ_{jk}^i are connection coefficients on \mathcal{S} . Since we use Euclidean space for \mathcal{S} , we have $\gamma_{jk}^i = 0$.

At this point, it is useful to discuss the Eulerian and Lagrangian viewpoints. Given the above definitions, any objects that are defined on \mathcal{B} are called Lagrangian or material, while the concepts defined on \mathcal{S} are called Eulerian or spatial. The Lagrangian and Eulerian point of view are equivalent, because anything that is defined in one can be transformed to the other. For cardiac tissue it is natural to use the Lagrangian framework. This has the advantage that we do not need convective derivatives in the description.

To model the cardiac microstructure, i.e., the fiber, sheet and normal direction, we will use *frame fields*, which are also called *vielbeins* in physics. Frame fields are a set of orthonormal vector fields. They span at each point of a manifold a basis for the tangent space. If G is the metric of our (material) manifold and $\{E_A\}_{A=1}^m$ the frame field, the orthonormality condition is

$$G(E_A, E_B) = \delta_{AB}. \quad (1)$$

The dual of the frame field is denoted E^A (with upper indices) and called the coframe field. It is defined to obey $E^A(E_B) = \delta_B^A$, such that it can be used to write the metric in the simple form

$$G = \sum_{A=1}^m E^A \otimes E^A. \quad (2)$$

We will denote the components of the frame field E_A in the coordinate basis by E_A^I and of the coframe field E^A by E_I^A .

2.2. Balance Equations

Although the bidomain and elasticity equations are well-known, we will still derive for consistency the equations of cardiac electromechanics here starting from basic continuum balance laws. This will allow us explicitly mention assumptions

and approximations made, and to emphasize that cardiac electromechanics is more than just the sum of bidomain and elasticity equations, giving rise to more complicated constitutive equations (such as deformation-dependent conduction tensors).

Our starting point are physical conservation laws: balance of charge in the intra- and extracellular domains, no accumulation of total charge, balance of momentum, and the dynamics of the internal variables (such as gating variables and ionic concentrations):

$$\frac{\partial Q_i}{\partial t} + \text{DIV } J_i = -I_{ion}, \quad (3a)$$

$$\frac{\partial Q_e}{\partial t} + \text{DIV } J_e = I_{ion}, \quad (3b)$$

$$\frac{\partial(Q_i + Q_e)}{\partial t} = 0, \quad (3c)$$

$$\rho_{\text{Ref}} A - \text{DIV } P - \rho_{\text{Ref}} B = 0, \quad (3d)$$

$$\frac{\partial \Gamma}{\partial t} = R, \quad (3e)$$

where Q_i and Q_e are the intra- and extracellular charge densities, J_i and J_e are the intra- and extracellular current densities, ρ_{Ref} is the reference mass density, A is the acceleration, P the first Piola-Kirchhoff stress tensor, B is the body force (e.g., gravity), Γ is a column matrix of the internal variables and R are their reaction rates. Note that all quantities live on the material manifold \mathcal{B} and DIV is the divergence operator on \mathcal{B} .

The assumptions in the bidomain formulation are the following. First, the cell membrane can be modeled as a capacitor: $Q_i - Q_e = 2C_m V_m$, where C_m is the capacitance per volume and $V_m = V_i - V_e$ the transmembrane voltage. Second, the intra- and extracellular space are ohmic conductors, with intra- and extracellular conductivities Σ_i and Σ_e . Thus we get the following set of equations:

$$\frac{\partial(C_m V_m)}{\partial t} + \text{DIV}(\Sigma_i \cdot \text{GRAD } V_m) + \text{DIV}(\Sigma_e \cdot \text{GRAD } V_e) = -I_{ion}, \quad (4a)$$

$$\text{DIV}(\Sigma_i \cdot \text{GRAD } V_m) + \text{DIV}((\Sigma_{i+e}) \cdot \text{GRAD } V_e) = 0, \quad (4b)$$

$$\frac{\partial \Gamma}{\partial t} = R, \quad (4c)$$

$$\rho_{\text{Ref}} A - \text{DIV } P - \rho_{\text{Ref}} B = 0. \quad (4d)$$

An assumption often made in cardiac mechanics is the neglect of the inertial term $\rho_{\text{Ref}} A$. This is justified because sound waves occur on a much faster time scale than the electrical waves in cardiac tissue: the ratio of the speed of sound to conduction velocity is around 25. This was also validated numerically in an electromechanical model of a 1D fiber (Whiteley et al., 2007).

2.3. Constitutive Equations

To close Equations (4) we need to specify constitutive equations for Σ_i , Σ_e , I_{ion} , R , and P . We will only consider the dependencies

as pointwise functions of material position X , transmembrane potential V_m , internal variables Γ and deformation C :

$$\Sigma_i = \hat{\Sigma}_i(X, C), \quad (5a)$$

$$\Sigma_e = \hat{\Sigma}_e(X, C), \quad (5b)$$

$$I_{ion} = \hat{I}_{ion}(X, V_m, \Gamma, C), \quad (5c)$$

$$R = \hat{R}(X, V_m, C) \quad (5d)$$

$$P = F\hat{S}(X, \Gamma, C). \quad (5e)$$

Instead of working with a function \hat{P} for the first Piola-Kirchhoff stress tensor, we directly work with a function \hat{S} for the second Piola-Kirchhoff stress tensor, because it is symmetric. It is also possible that the material capacitance depends on deformation, and therefore we write $C_m = \hat{C}_m(X, C)$. Based on the symmetries of the material we can deduce more specific representations for the scalar (\hat{I}_{ion} , \hat{R} , and \hat{C}_m) and symmetric second order tensor functions ($\hat{\Sigma}_i$, $\hat{\Sigma}_e$, \hat{S}). Because of the specific microstructure of cardiac tissue, we only focus on orthotropic materials, but more general symmetries based on crystal groups are possible (Smith, 2012). For the following we will use the notation $\{E_A\}_{A \in \{F, S, N\}}$ for the local fiber, sheet and sheet normal directions.

Let us start with the scalar functions. It can be shown (Itskov, 2013) that every scalar-valued function of a symmetric rank-2 tensor M , such as the deformation tensor C , the second Piola-Kirchhoff stress tensor S and the conduction tensors Σ_i , Σ_e , which is invariant under orthotropic symmetries can be written as a function of the seven invariants $\{M_{FF}, M_{SS}, M_{NN}, (M_{FS})^2, (M_{FN})^2, (M_{SN})^2, M_{FS}M_{SN}M_{NF}\}$. If $\det(M) = 1$, (e.g., when M is the deformation tensor of an incompressible material), these seven invariants are not independent anymore and we can leave out the last one. In that case our scalar constitutive equations would be a function of the six invariants $\{M_{FF}, M_{SS}, M_{NN}, (M_{FS})^2, (M_{FN})^2, (M_{SN})^2\}$. Often \hat{I}_{ion} and \hat{R} are taken to be a function of the fiber stretch $\lambda = \sqrt{C_{FF}}$ only, see for example Niederer et al. (2006) and Panfilov et al. (2007).

Orthotropic tensor-valued functions T of a symmetric tensor M can be shown to be of the form (Itskov, 2013)

$$\begin{aligned} \hat{T}(M) = \sum_{A \in \{F, S, N\}} & \left[\hat{\alpha}_A (E_A \otimes E_A) \right. \\ & + \frac{\hat{\beta}_A}{2} (M \cdot E_A \otimes E_A + E_A \otimes E_A \cdot M) \\ & + \frac{\hat{\gamma}_A}{2} (M^2 \cdot E_A \otimes E_A + E_A \otimes E_A \cdot M^2) \\ & + \frac{\hat{\delta}_A}{2} (M \cdot E_A \otimes E_A - E_A \otimes E_A \cdot M) \\ & \left. + \frac{\hat{\epsilon}_A}{2} (M^2 \cdot E_A \otimes E_A - E_A \otimes E_A \cdot M^2) \right], \end{aligned} \quad (6)$$

where $\hat{\alpha}$, $\hat{\beta}$, $\hat{\gamma}$, $\hat{\delta}$, and $\hat{\epsilon}$ are now scalar-valued functions of M . Note that for $\hat{T}(M)$ symmetric $\hat{\delta}_A = \hat{\epsilon}_A = 0$ while for $\hat{T}(M)$ antisymmetric $\hat{\alpha}_A = \hat{\beta}_A = \hat{\gamma}_A = 0$.

When we write out this expression in components of the E_A frame ($A, B \in \{F, S, N\}$, no summation implied) we get:

$$\begin{aligned} \hat{T}_{AB}(M) = & \frac{\hat{\alpha}_A + \hat{\alpha}_B}{2} \delta_{AB} + \frac{\hat{\beta}_A + \hat{\beta}_B + \hat{\delta}_A - \hat{\delta}_B}{2} M_{AB} \\ & + \frac{\hat{\gamma}_A + \hat{\gamma}_B + \hat{\epsilon}_A - \hat{\epsilon}_B}{2} (M^2)_{AB}. \end{aligned} \quad (7)$$

The second Piola-Kirchhoff stress tensor S is symmetric and in the case that it is hyperelastic (such that $\hat{S}^H(C) = 2 \frac{\partial \hat{\psi}}{\partial C_{IJ}}$, where $\hat{\psi}$ is a function of the invariants), the constitutive equation simplifies to

$$\begin{aligned} \hat{S}(C) = \sum_{A \in \{F, S, N\}} & \left[\hat{\alpha}_A (E_A \otimes E_A) + \frac{\hat{\beta}_A}{2} (C \cdot E_A \otimes E_A \right. \\ & \left. + E_A \otimes E_A \cdot C) \right] + \hat{\gamma} C^2. \end{aligned} \quad (8)$$

For ventricular cardiac tissue, the Guccione (Guccione et al., 1995) and Holzapfel-Ogden (Holzapfel and Ogden, 2009) constitutive equations are popular choices.

Throughout the literature on cardiac electromechanical modeling, several deformation-dependent conduction tensors have been proposed. The simplest form is obtained by making the conduction coefficients Σ_A dependent on the stretch along the principal material directions: with $\lambda_A = \sqrt{C(E_A, E_A)}$,

$$\hat{\Sigma}(C) = \sum_{A \in \{F, S, N\}} \hat{\Sigma}_A(\lambda_A) E_A \otimes E_A \quad (9)$$

Examples for these are $\hat{\Sigma}_A(\lambda_A) = \Sigma_A$, i.e., deformation-independent or “gap-junction based” conduction (Bakir and Dokos, 2015) or $\hat{\Sigma}_A(\lambda_A) = \frac{\Sigma_A}{\lambda_A^2}$ (Colli Franzone et al., 2016). Yet another form for the conduction tensor can be found in Bakir and Dokos (2015), which they call “spatially based” conduction:

$$\hat{\Sigma}(C) = JU^{-1} \cdot \left(\sum_{A \in \{F, S, N\}} \Sigma_A(\lambda_A) E_A \otimes E_A \right) \cdot U^{-T}, \quad (10)$$

where U is the right stretch tensor, i.e., $U = \sqrt{C}$. A related form is (Sachse, 2004):

$$\hat{\Sigma}(C) = W \cdot \left(\sum_{A \in \{F, S, N\}} \Sigma_A(\lambda_A) E_A \otimes E_A \right) \cdot W^T, \quad (11)$$

where $W = U^{-1} (\mathbb{1} + \theta(U - \mathbb{1}))$ and $\theta \in [0, 1]$ is a parameter which reduces this conduction tensor to the “spatial based” conduction for $\theta = 0$ (apart from the Jacobian factor) and to a “gap-junction based” conduction for $\theta = 1$.

In Göktepe and Kuhl (2010) and Göktepe et al. (2013) the following transversely isotropic form

$$\hat{\Sigma}(C) = \Sigma_{iso} C^{-1} + \Sigma_{ani} E_F \otimes E_F \quad (12)$$

was used and in Plank et al. (2013):

$$\hat{\Sigma}(C) = \left(\sum_{A \in \{F, S, N\}} \Sigma_A E_A \otimes E^A \right) \cdot C^{-1}. \quad (13)$$

This variety of deformation-dependent conduction tensors is mostly a consequence of the assumptions that were made about the conduction coefficients, for example one assumes that the conduction coefficients are constant in the spatial or in the material frame. However, nothing says a priori if these should even be constant. So to have realistic deformation-dependent conduction tensors relationships, the conduction coefficients should be based on measurements with different deformations.

2.4. Variational Formulation

In view of the time-integration methods which will be presented in section 4.1, let us split R in fast processes (to be treated implicitly) and slow processes: $R = R_I + R_E$. Furthermore, let P_{appl} denote the applied pressure on the pressure boundary of the deformation ϕ (e.g., the fluid pressure at the endocardial surfaces). Writing the fast processes on the left-hand side and the slow processes at the right-hand side, the weak or variational form for electromechanics can be written as: find V_m, V_e, Γ, ϕ such that

$$\int_B \delta V_m \frac{\partial V_m}{\partial t} dV + \int_B \delta V_m|_I (\Sigma_i)^{IJ} (V_m|_J + V_e|_J) dV = - \int_B \delta V_m I_{ion} dV \quad (14a)$$

$$\int_B \delta V_e|_I ((\Sigma_i)^{IJ} V_m|_J + (\Sigma_{i+e})^{IJ} V_e|_J) dV = 0 \quad (14b)$$

$$\int_B \delta \Gamma \left(\frac{\partial \Gamma}{\partial t} - R_I \right) dV = \int_B \delta \Gamma R_E dV \quad (14c)$$

$$\int_B \delta \phi^i|_I P_i^I dV + \int_{\partial_N B} \delta \phi^i P_{appl} I (F^{-1})^I_i N_I dS = 0, \quad (14d)$$

for all test functions $\delta V_m, \delta V_e, \delta \Gamma, \delta \phi$. The notation $|_I$ was introduced for the I th component of the covariant derivative, i.e., $\delta V_e|_I = \frac{\partial V_m}{\partial X^I}$ and $\delta \phi^i|_I = \frac{\partial(\delta \phi^i)}{\partial X^I} + \gamma_{jk}^i \delta \phi^j F_I^k$ (again, for Euclidean S the connection γ vanishes).

Note that we can write any left-hand side of (14) in the following form:

$$\int_B (v \cdot f_0 + \nabla v : f_1) dV + \int_{\partial_N B} v \cdot g_0 dS \quad (15)$$

where v represents any of the test functions and f_0, f_1 , and g_0 are general functions of V_m, V_e, Γ , and ϕ , their gradients and time derivatives, time and spatial coordinates. More specifically, we can summarize all the fast physics by pointwise functions in the following table:

	f_0	f_1	g_0
V_m	$\frac{\partial V_m}{\partial t}$	$\Sigma_i \cdot \nabla V_m + \Sigma_i \cdot \nabla V_e$	
V_e		$\Sigma_i \cdot \nabla V_m + (\Sigma_{i+e}) \cdot \nabla V_e$	
Γ	$\frac{\partial \Gamma}{\partial t} - R_I$		
ϕ		P	$P_{appl} J F^{-T} \cdot N$

(16)

For implicit time integration we will also need the Jacobian of the left-hand side. Its action on the increments $\Delta V_m, \Delta V_e, \Delta \Gamma$, and $\Delta \phi$ is given by

$$\int_B \delta V_m \gamma \Delta V_m dV + \int_B \delta V_m|_I (\Sigma_i)^{IJ} \Delta V_m|_J dV \quad (17a)$$

$$\int_B \delta V_m|_I (\Sigma_i)^{IJ} \Delta V_e|_J dV \quad (17b)$$

$$\int_B \delta V_e|_I (\Sigma_i)^{IJ} \Delta V_m|_J dV \quad (17c)$$

$$\int_B \delta V_e|_I (\Sigma_{i+e})^{IJ} \Delta V_e|_J dV \quad (17d)$$

$$\int_B \delta \Gamma \left(\gamma - \frac{\partial R_I}{\partial \Gamma} \right) \Delta \Gamma dV \quad (17e)$$

$$\int_B \delta \phi^i|_I A_i^{IJ} \Delta \phi^j|_J dV + \int_{\partial_N B} \delta \phi^i P_{appl} B_{ij}^J \Delta \phi^j|_J dS \quad (17f)$$

where γ is the shift factor determined by the numerical integration scheme (for example, for backward Euler with time step h , $\gamma = h^{-1}$) and

$$B_{ij}^J = \frac{\partial (J (F^{-1})^I_i N_I)}{\partial F_j^J} = J N_I \left((F^{-1})^I_i (F^{-1})^J_j - (F^{-1})^I_j (F^{-1})^J_i \right), \quad (18)$$

and

$$A_i^{IJ} = \frac{\partial P_i^I}{\partial F_j^J} \quad (19)$$

is called the first elasticity tensor (Marsden and Hughes, 1994).

The expressions (17) can generally be written as

$$\int_B [v^T \nabla v^T] \begin{bmatrix} f_{0,0} & f_{0,1} \\ f_{1,0} & f_{1,1} \end{bmatrix} \begin{bmatrix} w \\ \nabla w \end{bmatrix} dV + \int_{\partial_N B} [v^T] \begin{bmatrix} g_{0,0} & g_{0,1} \end{bmatrix} \begin{bmatrix} w \\ \nabla w \end{bmatrix} dS \quad (20)$$

and the pointwise Jacobians can be summarized as

	$f_{0,0}$	$f_{1,1}$	$g_{0,1}$
(V_m, V_m)	γ	Σ_i	
(V_m, V_e)		Σ_i	
(V_e, V_m)		Σ_i	
(V_e, V_e)		Σ_{i+e}	
(Γ, Γ)	$\gamma - \frac{\partial R_I}{\partial \Gamma}$		
(ϕ, ϕ)		A	$P_{appl} B$

(21)

where for example (V_m, V_e) refers to the derivative of the weak equation for V_m w.r.t. V_e .

3. DISCRETIZATION

In this section we apply standard methods to express the variational equations in a finite element basis, to obtain a

large non-linear system to solve instead of continuum partial differential equations.

We will use the finite element method (Ciarlet, 2002; Brenner and Scott, 2007; Zienkiewicz et al., 2013) to spatially discretize the weak forms (14). Let the manifold \mathcal{B} be triangulated into E m -simplices $\{K^e\}_{e=1}^E$ (cells/elements), each diffeomorphic to the standard m -simplex \hat{K} (with coordinates $\xi^{\hat{I}}$): for each e there is a coordinate map $\hat{X}^e: \hat{K} \rightarrow K^e$ for which their element Jacobians $(J^e)_I^I = \frac{\partial \hat{X}^I}{\partial \xi^I}$ and their inverse exist (and are continuous). If we also choose a function space P and a basis for the dual space Σ over each element, the triple (K, P, Σ) defines the finite element (Ciarlet, 2002). Here we will only use 1st order Lagrange elements (Brenner and Scott, 2007). Let $\{\varphi_p\}_{p=1}^{\dim P}$ denote the basis functions for P and let $\{\xi_q\}_{q=1}^Q$ and $\{w_q\}_{q=1}^Q$ be the quadrature points of a quadrature rule with Q quadrature points (e.g., Gauss-Jacobi in the case of simplices Karniadakis and Sherwin, 2013). Then we can define the element basis evaluation, derivative and integration matrices as $(B^e)_{qp} = \varphi_p(\xi_q)$, $(D_I^e)_{qp} = \frac{\partial \varphi_p}{\partial \xi^I}(\xi_q) ((J^e)^{-1})_I^I$ and $(W^e)_{qp} = \delta_{qp} w_q \det(J^e)$

Following (Brown, 2010; Knepley et al., 2013) we discretize the volume terms

$$\int_{\mathcal{B}} (v \cdot f_0 + \nabla v : f_1) dV \quad (22)$$

as

$$\sum_e \mathcal{E}_e^T \left[(B^e)^T W^e \Lambda^e(f_0) + \sum_I (D_I^e)^T W^e \Lambda^e(f_1^I) \right], \quad (23)$$

where \mathcal{E}_e is the element restriction operator and Λ^e transforms a function into function evaluations at the quadrature points. Note that evaluation of a field u at the quadrature points are evaluated as $u^e = B^e \mathcal{E}_e u$ and their derivatives as $\nabla_I u^e = D_I^e \mathcal{E}_e u$.

The boundary integrals

$$\int_{\partial_N \mathcal{B}} v \cdot g_0 dS \quad (24)$$

are discretized as

$$\sum_f \mathcal{E}_{e(f)}^T \left[(B^{e(f)})^T W^f \Lambda^{e(f)}(g_0) \right], \quad (25)$$

where $e(f)$ refers to the neighboring element of f , i.e., we evaluate at the quadrature points of the face using the neighboring element's basis functions and field coefficients.

4. ALGORITHMS

In this section we present IMEX integration schemes, the resulting non-linear equations and approaches to solve them numerically for the specific structure of the electromechanical equations.

4.1. Time Integration Using IMEX Schemes

For systems that have multiple time scales that are well-separated, we have to choose a time scale that we are interested in. In studying the long term or slowly varying behavior, the fast transient processes don't need to be fully resolved, as these decay rapidly. These systems are called stiff (see Söderlind et al., 2015 for a discussion on stiffness). Note that in discretized PDEs, the fastest time scale often comes in the form of a Courant-Friedrichs-Levy limit (Courant et al., 1928), making it mesh-dependent.

Explicit schemes require the time step to be of the same order as the fastest process for stability, so they are very inefficient for stiff systems. Implicit schemes can step over those fast processes, but the downside is that they produce large fully coupled non-linear systems. Implicit-Explicit (IMEX) schemes combine the best of both worlds: they integrate the fast processes implicitly and the slow processes explicitly. A class of IMEX methods are Additive Runge-Kutta Implicit-Explicit (ARKIMEX) schemes (Ascher et al., 1997; Kennedy and Carpenter, 2001; Giraldo et al., 2013). They combine two s-stage methods (ERK and (ES)DIRK), summarized by two Butcher tableaus (Butcher, 2016)

$$\begin{array}{c|ccc} c_1^E & a_{11}^E & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ c_s^E & a_{s1}^E & \cdots & 0 \\ \hline & b_1^E & \cdots & b_s^E \end{array} \quad \begin{array}{c|ccc} c_1^I & a_{11}^I & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ c_s^I & a_{s1}^I & \cdots & a_{ss}^I \\ \hline & b_1^I & \cdots & b_s^I \end{array}, \quad (26)$$

additively to integrate equations of the following form

$$M \dot{y} = f^I(y, t) + f^E(y, t), \quad (27)$$

where $y: I \rightarrow \mathbb{R}^N$ describes the evolution of the discretized state, f^I and f^E are resp. the implicitly and the explicitly treated functions and M is a mass matrix. The implicit function contains the fast or stiff physics, whereas the explicit function contains the slow or non-stiff physics. Often f^I is linear and f^E non-linear. The i -th stage value Y_i can then be computed as

$$Y_i = y_n + h \sum_{j=1}^{i-1} a_{ij}^E \dot{Y}_j^E + h \sum_{j=1}^i a_{ij}^I \dot{Y}_j^I, \quad (28)$$

where the implicit and explicit stage derivatives are given by $\dot{Y}_i^I = M^{-1} f^I(Y_i, t_n + c_i h)$ and $\dot{Y}_i^E = M^{-1} f^E(Y_i, t_n + c_i h)$. The difference between both terms is that the stage Y_i depends on only previous stages for the explicit part, but also on the current stage for the implicit part. The numerical constants a_{ij}^I , a_{ij}^E follow from the chosen integration scheme, see the Butcher tableaus (26).

After rearranging, Equation (28) produces a non-linear equation in Y_i , if the $a_{ii}^I \neq 0$:

$$M \gamma (Y_i - Z_i) - f^I(Y_i, t_n + c_i h) = 0, \quad (29)$$

where γ is the shift factor determined by the numerical integration scheme (for example, for backward Euler with time step h , $\gamma = h^{-1}$). The Jacobian for this equation is

$$\gamma M - \frac{\partial f^I}{\partial y}(Y_i, t_n + c_i h) \quad (30)$$

and is used while iteratively solving Equation (29) for Y_i . Thereafter, the implicit stage derivative can be simply found as

$$\dot{Y}_i^I = \gamma(Y_i - Z_i) \quad (31)$$

and the explicit stage derivative by evaluating the explicit function

$$\dot{Y}_i^E = M^{-1}f^E(Y_i, t_n + c_i h). \quad (32)$$

The solution at the next time step is then calculated as

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i^E \dot{Y}_i^E + h \sum_{j=1}^s b_j^I \dot{Y}_j^I. \quad (33)$$

Note that if $f^I = 0$ we have a purely explicit scheme and if $f^E = 0$ we have a purely implicit scheme. In order to avoid the need to invert M , we will only use schemes for which $a_{si}^I = b_i^I$ and $a_{si}^E = b_i^E$, the so-called globally stiffly accurate schemes (Boscario et al., 2013). Then, the completion step (33) can be skipped. For a more thorough discussion on the technical aspects, we refer to Kennedy and Carpenter (2001). In the context of electrophysiology they were previously applied only to single cell models, where they have been shown to outperform other integration schemes (Spiteri and Dean, 2008).

4.2. Non-linear Solvers

The IMEX schemes allow us to put some of the complicated non-linear dependencies in the right-hand sides, making the implicit solve easier. If we make the following assumptions, we can essentially solve the whole non-linear system by solving each subproblem one after another: the ionic current, the stretch-dependent terms in the cell models and dependence of the tension variables on Ca_i or V_m must be in the RHS. Now we can solve for the stage values by doing the following: first solve the active tension internal variable equations, then solve the mechanical equations (14d), then solve the bidomain equations ((14a) and (14b)) together and finally solve the electrophysiological internal variable equations (14c). This approach is nothing more than the non-linear Gauss-Seidel method applied to the fields:

Algorithm 1 Nonlinear Gauss-Seidel

```

Given initial  $u = (u_1, \dots, u_n)^T$ 
for  $k = 1, \dots, n$  do
  Solve  $F_k(u_1^*, \dots, u_k^*, \dots, u_n) = 0$  for  $u_k^*$ 
end for

```

During this process, we solve the bidomain and, if possible, the implicit internal variables equations with a linear solver (to be specified below), while we solve the non-linear mechanical equations with Newton's method. If for some reason some of the above assumptions do not hold and coupling between variables is strong enough, more Gauss-Seidel sweeps are done to converge. Alternatively, one could use the above algorithm as a non-linear preconditioner (Liu and Keyes, 2015).

4.3. Linear Solvers and Preconditioners

4.3.1. Bidomain

We solve the discretized bidomain equations with conjugate gradients preconditioned by block preconditioners (Sundnes et al., 2002; Pennacchio and Simoncini, 2009; Bernabeu et al., 2010; Pavarino and Scacchi, 2011). For this we use PETSc's FieldSplit preconditioner, allowing us to flexibly choose between different strategies (Brown et al., 2012) from the command line. Both blocks are preconditioned with one V-cycle of PETSc's native algebraic multigrid preconditioner (GAMG). If no Dirichlet boundary conditions are given for the extracellular voltage, we also provide the constant nullspace vector to the respective block solve.

4.3.2. Mechanics

We solve the linearized elasticity equations arising from Newton's method with conjugate gradients, preconditioned with PETSc's algebraic multigrid preconditioner. The difference here with previous work (Franzone et al., 2015; Gurev et al., 2015; Augustin et al., 2016) is that this algebraic multigrid preconditioner uses smoothed aggregation (Vaněk et al., 1996), which is more efficient for elasticity problems (Van et al., 2001; Adams, 2002). We provide the rigid body modes to PETSc's GAMG preconditioner to obtain more accurate coarse spaces, resulting in a significant drop in iterations. Here we use a full multigrid cycle as this also helps in lowering the number of iterations of the linear solver at the expense of only a small percentage more work than a single V-cycle.

4.3.3. Internal Variables

As the internal variables on different points are completely decoupled these can be solved easily as small linear systems. Very often these systems are even diagonal, for example when most of the stiffness comes from the gating variables.

5. IMPLEMENTATION: GEMS

5.1. Source Code in C Using PETSc

We implemented our code in C using the PETSc library (Balay et al., 1997, 2016a,b). This allows us to have a large choice of scalable and efficient algorithms and data structures for the solution of time-dependent PDE's, which can be easily changed or finetuned through command line options. By using PETSc's unstructured mesh data structure, we can easily read and write common mesh formats, (re)distribute meshes and associated data and we have access to powerful solvers which need access to mesh and field information (e.g., multigrid and block preconditioners). More specifically, we used DMPlex (Isaac and Knepley, 2015; Knepley et al., 2015; Lange et al., 2015) for mesh management and PetscFE for finite element technology, TS (Abhyankar, 2014) for time stepping, SNES for non-linear solvers and KSP/PC for linear solvers and preconditioners. Input and output routines are coupled to PETSc. Meshes can be read in through DMPlex if it is of the ExodusII (Schoof and Yarberr, 1994), Gmsh (Geuzaine and Remacle, 2009), CGNS (Poirier et al., 1998), MED (Open CASCADE, 2017), Fluent Case (Fluent, 2006), or PLY (Wikipedia, 2017) file format. Alternatively, meshes can also

be created by giving the vertex numbers per cell and vertex coordinates. Output can be generated using the builtin PETSc viewers. For example, DM (mesh) and Vec (representing discrete fields) objects can be stored as HDF5 (The HDF Group, 1997–2017) data, which can be read by ParaView (Ayachit, 2015) or VisIt (Childs et al., 2012) with XDMF metadata (Kitware, 2017). The extensible nature of PETSc also makes it possible to implement new solvers and use them through PETSc. This way we implemented a SNES solver called SNESFieldSplit, which is the non-linear block Gauss-Seidel solver we discussed in 4.2. Once this solver knows about the field layout and the equations per field through the DM, it can automatically do the subsolves. This is the non-linear equivalent to PCFieldSplit (Brown et al., 2012), already in PETSc.

5.2. Main GEMS Classes and Usage

The most important part of our GEMS library is the *GEMSModel* class. It is responsible for providing all the model-dependent information such as pointwise residuals and Jacobians, discretizations, null spaces, and initial guess/conditions to the appropriate PETSc classes. Current subclasses include *GEMSModelMonodomain*, *GEMSModelBidomain*, *GEMSModelElasticity*, *GEMSModelElectromechanics* (combining monodomain and quasi-static elasticity), and *GEMSModelFibres* (to create rule-based fiber directions based on solving Laplace equations, following Bayer et al., 2012).

Typical usage for a non-linear problem is illustrated in 1. Note that nothing should be done extra to run simulations in different dimensions besides changing the mesh, which can be as simple as just changing the filename of the mesh. The ...FromOptions(...) functions are meant to be configured from the command line or options file. For example, if the *GEMSModel* should be changed to *GEMSModelMonodomain*, the option *-gemsmodel_type monodomain* would be added to the command line or options file.

Listing 1 | Typical usage of the *GEMSModel* class

```
MPI_Comm comm;
SNES      snes;
DM         dm;
Vec        u;
GEMSModel model;

/* Initialize GEMS, PETSc, MPI, read options */
GEMSInitialize(&argc, &argv, NULL, help);
comm = PETSC_COMM_WORLD;

/* Create a DMplex using, e.g., DMplexCreateFromFile() */
DMplexCreate...(comm, ..., &dm);

/* Create and configure a GEMSModel */
GEMSModelCreate(comm, &model);
GEMSModelSetFromOptions(model);
/* Set model-specific discretizations and equations in the DM */
```

```
GEMSModelSetUpDiscretization(model, dm);
/* Create model-specific near-null space (this is used by GAMG) */
GEMSModelCreateNearNullSpace(model, dm, NULL);

/* Create and initialize the solution vector */
DMCreateGlobalVector(dm, &u);
PetscObjectSetName((PetscObject)u, "solution");
ModelInitializeSolutionVector(model, dm, u);

/* Use DMplex's internal FEM routines */
DMSNESSetBoundaryLocal(dm,
    DMplexSNESComputeBoundaryFEM, NULL);
DMSNESSetFunctionLocal(dm,
    DMplexSNESComputeResidualFEM, NULL);
DMSNESSetJacobianLocal(dm,
    DMplexSNESComputeJacobianFEM, NULL);

/* Create and configure the nonlinear solver and solve */
SNESCreate(comm, &snes);
SNESSetDM(snes, dm);
SNESSetFromOptions(snes);
SNESolve(snes, NULL, u);

/* View the mesh */
DMViewFromOptions(dm, NULL, "-dm_view");

/* View the solution */
VecViewFromOptions(u, NULL, "-sol_vec_view");

/* Clean up */
SNESDestroy(&snes);
VecDestroy(&u);
ModelDestroy(&model);
DMDestroy(&dm);
GEMSFinalize();
```

Further we have a class for the electrophysiological 0D cell models called *GEMSCellModel*. Its only function is to give the pointwise implicit and explicit functions, Jacobian and initial conditions. Currently implemented cell models include FitzHugh-Nagumo (FitzHugh, 1961; Nagumo et al., 1962) and Ten Tusscher-Panfilov 2006 (ten Tusscher and Panfilov, 2006) models.

5.3. Comparison to Other Cardiac Solvers

One of the main features of GEMS is, that it uses PETSc (and other third party packages it interfaces) as much as possible and not just as a linear algebra solver. In particular it uses the DM object prominently, which makes it easy to input/output meshes and field data in various formats, feed field and mesh data to

various advanced (non)linear, often consisting of combinations of subsolvers, etc (for example, the block preconditioners for bidomain or incompressible elasticity in which each field has a different preconditioner and linear iterative solver). These solvers (and their subsolvers) can then be configured just from command line options, without recompiling. Thus it strives for maximal flexibility and easy experimentation. Other cardiac solvers such as Chaste (Mirams et al., 2013) or Continuity (Continuity, 2018) have already existed for many years and have functionalities such as reading generic cell models through CellML and solving mechanics. But the typical approach to electromechanics is first order operator splitting with separate codes for mechanics and electrophysiology. Our library was built with a flexible approach to coupling between different physics from the beginning. To specify a problem we start with a coupled set of equations (defined by pointwise residuals, right hand sides and Jacobians) and through command line options we can configure the solvers. This makes experimentation with different combinations of solvers a whole lot easier and also makes it possible to use higher order integration schemes.

6. NUMERICAL RESULTS

6.1. Electrophysiology

As a first test we did the benchmark for electrophysiology with the cardiac monodomain equations as described in Niederer et al. (2011), with the suggested spatial resolutions of 0.5, 0.2, and 0.1 mm (using linear tetrahedral elements) and temporal resolutions of 0.05, 0.01, and 0.005 ms. We did the benchmark of propagation in a 3D slab with three different integration schemes: with FBE111 (forward-backward Euler), the ARS222 (Ascher et al., 1997), and the BPR353 schemes (Boscario et al., 2013)

(the numbers in the names of these integration scheme names reflect the number of explicit and implicit stages and the order of accuracy). As an extra, we also ran the benchmark using a large time step of 0.5 ms at a spatial resolution of 0.1 mm, to showcase the stability and temporal convergence of the used methods. The internal variables were stored at the quadrature points. In **Figure 1** we display the activation times along the diagonal of the bar geometry. We see that increasing spatial and temporal resolutions have opposite effects on arrival times: increasing spatial resolution raises the arrival time, while increasing the temporal resolution lowers the arrival time. The faster convergence rate of the arrival time for higher order time integration is also noticeable. For example, for the BPR353 scheme the arrival times for the time steps of 0.05, 0.01, and 0.005 ms are almost indistinguishable. In Niederer et al. (2011) different codes were found to have arrival times between 37.8 and 48.7 ms at the highest spatial and temporal resolutions. Our arrival times are within those bounds at these highest resolutions. (It is inevitable that at lower resolutions the arrival time will deviate more.) Execution times for the simulations can be found in **Table 1**.

6.2. Electromechanics

At this stage of development of our package we decided just to illustrate the solution of electromechanical equations using the most simple tools. The comparison of various integration methods and constitutive relations will be done at a later stage. As an illustration for the fully coupled electromechanical equations we simulated the contraction of an idealized biventricular geometry that was stimulated at the apex. The mesh for this geometry was created using Gmsh (Geuzaine and Remacle, 2009) with a resolution of 0.2 mm resulting in a tetrahedral

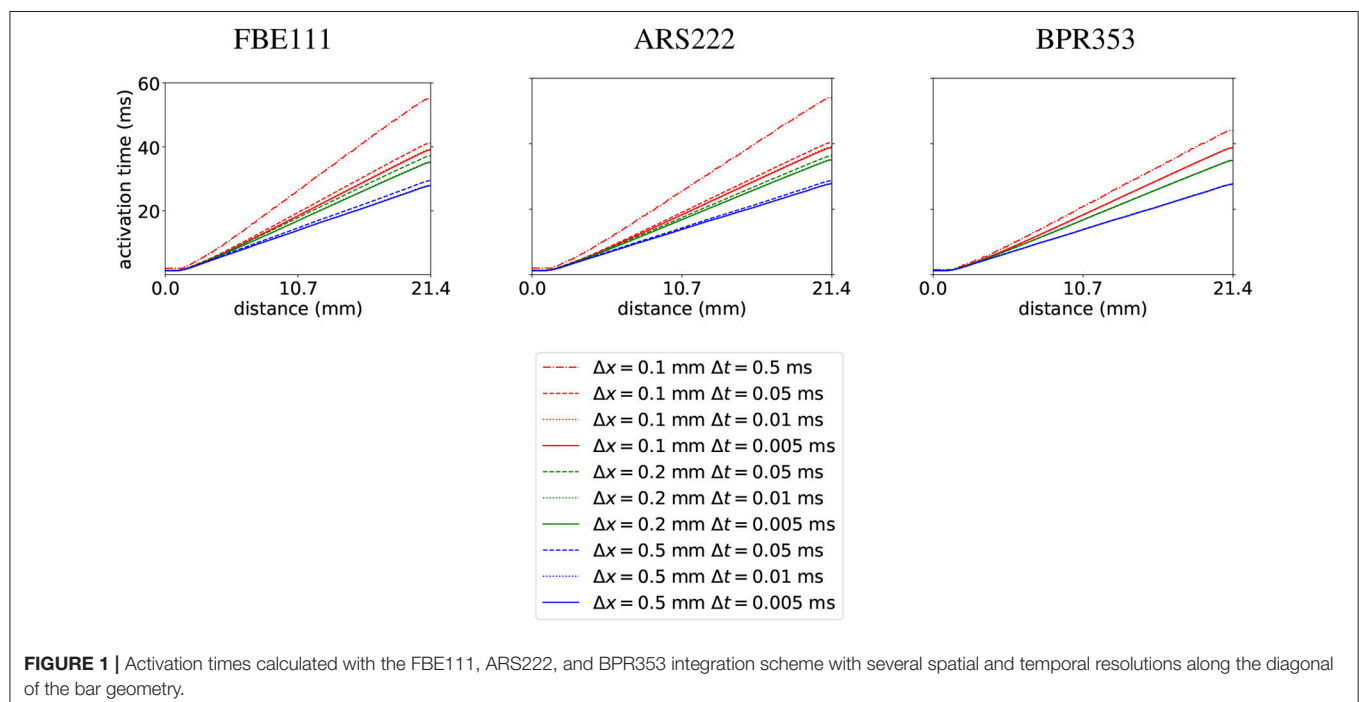


Table 1 | Execution times for Niederer's electrophysiology benchmark.

Scheme	Δx (mm)	Δt (ms)	Execution time (s)
FBE111	0.1	0.5	$4.20 \cdot 10^3$
FBE111	0.1	0.05	$1.27 \cdot 10^4$
FBE111	0.1	0.01	$6.34 \cdot 10^4$
FBE111	0.1	0.005	$1.24 \cdot 10^5$
FBE111	0.2	0.05	$1.81 \cdot 10^3$
FBE111	0.2	0.01	$9.00 \cdot 10^3$
FBE111	0.2	0.005	$1.82 \cdot 10^4$
FBE111	0.5	0.05	$2.94 \cdot 10^2$
FBE111	0.5	0.01	$1.48 \cdot 10^3$
FBE111	0.5	0.005	$3.10 \cdot 10^3$
ARS222	0.1	0.5	$5.40 \cdot 10^3$
ARS222	0.1	0.05	$2.51 \cdot 10^4$
ARS222	0.1	0.01	$1.25 \cdot 10^5$
ARS222	0.1	0.005	$2.48 \cdot 10^5$
ARS222	0.2	0.05	$3.48 \cdot 10^3$
ARS222	0.2	0.01	$1.77 \cdot 10^4$
ARS222	0.2	0.005	$3.44 \cdot 10^4$
ARS222	0.5	0.05	$4.30 \cdot 10^2$
ARS222	0.5	0.01	$2.19 \cdot 10^3$
ARS222	0.5	0.005	$4.40 \cdot 10^3$
BPR353	0.1	0.5	$1.10 \cdot 10^4$
BPR353	0.1	0.05	$5.27 \cdot 10^4$
BPR353	0.1	0.01	$2.59 \cdot 10^5$
BPR353	0.1	0.005	$5.22 \cdot 10^5$
BPR353	0.2	0.05	$7.28 \cdot 10^3$
BPR353	0.2	0.01	$3.57 \cdot 10^4$
BPR353	0.2	0.005	$7.12 \cdot 10^4$
BPR353	0.5	0.05	$7.45 \cdot 10^2$
BPR353	0.5	0.01	$3.67 \cdot 10^3$
BPR353	0.5	0.005	$7.15 \cdot 10^3$

Simulations were run on 32 nodes of Intel E5-2670 CPUs, using 1 core per node. See section 6.1 for details.

mesh consisting of 1529230 cells and 312888 vertices. We used the algorithm from Bayer et al. (2012) to generate myofiber orientations. The fiber angle varied from -45° (epi) to 75° (endo). We used the monodomain formulation and the TNNP06 (ten Tusscher and Panfilov, 2006) model for electrophysiology, with the same parameters as in Niederer et al. (2011). For the passive hyperelastic equations we used the Guccione constitutive equations (Guccione et al., 1995), where a penalty term $\kappa/2(J - 1)^2$ was added to the strain energy and for the active tension generation we used the Niederer-Hunter-Smith model (Niederer et al., 2006). Parameters were taken from Keldermann et al. (2010) and κ was taken as 350 kPa. Here we used a timestep of 0.5 ms with the FBE111 scheme and we used linear elements for the transmembrane voltage and deformations, while the internal variables were stored at the quadrature points. The resulting activation and contraction sequence can be seen in **Figure 2**. The simulation took 7.5 h on 32 nodes of Intel E5-2670 CPUs, using 1 core per node. The electromechanical testing will be continued in subsequent studies.

7. DISCUSSION AND OUTLOOK

In this paper we presented an overview of the methodology used in cardiac electromechanics and our numerical approach to these challenging problems. In particular, in section 2 we presented a short derivation of the main equations of electromechanics from basic principles (i.e., geometry and balance equations) in strong and weak form. We discussed constitutive equations to close these equations and clearly list all assumptions made. We derived a general representation of a deformation-dependent conduction tensor, assuming orthotropic symmetry and pointwise dependence on deformation and showed that previous deformation-dependent conduction tensors found in literature are all special cases of this. Note however, that the scalar functions in this representation still need to be determined from experiment. In section 3 we applied standard finite element methods to express the variational equations in a finite basis, which can then be solved by the numerical methods in section 4. There we discussed additive implicit-explicit Runge-Kutta time integration methods and how with appropriate partitioning of fast and slow physics the non-linear implicit equations can be solved more easily by solving smaller problems belonging to different fields one after another. Efficient (non-)linear solvers for these problems were also discussed. Further we reviewed the structure and possibilities of the GEMS library in section 5 and how PETSc gives us a wide range of tools to solve our PDE's, including meshes, I/O and solvers. In section 6 we presented some numerical results as verification and illustration of the GEMS library.

Our main conclusion is that additive implicit-explicit Runge-Kutta time integration methods, combining the advantages of implicit and explicit integration, work very well for electromechanical problems. This method allows larger time steps, with limited complication of Jacobians and non-linear solves. Our numerical implementation uses the PETSc library extensively, which gave us access to powerful and scalable mesh management, time stepping and (non)linear solvers which may need mesh and field information. One of the things which could be further researched is whether we can get much advantage of anisotropic mesh adaptation through the PRaGMaTic library (Rokos and Gorman, 2013), which has been recently interfaced to PETSc (Barral et al., 2016). This could also be used to build mesh hierarchies in a geometric multigrid approach.

The GEMS package is still in the process of further development. Although the user can access and set all solver options through the command line, a graphical user interface may be desirable in the future, both for input and visualization. Regarding modeling, we currently hard-coded two cell models (FHN and TP06) and foresee to import more models of cardiac electrophysiology in a semi-automated way via the CellML repository (www.cellml.org). We are currently using pressure boundary conditions on the endocardial surface, which can be extended with physical models for circulation and valve action.

Our method has been designed to enable strong coupling between the electrical and mechanical subsystems at every time step of the simulation, and at the same high spatial resolution, both for the electrical and mechanical equations. One possible

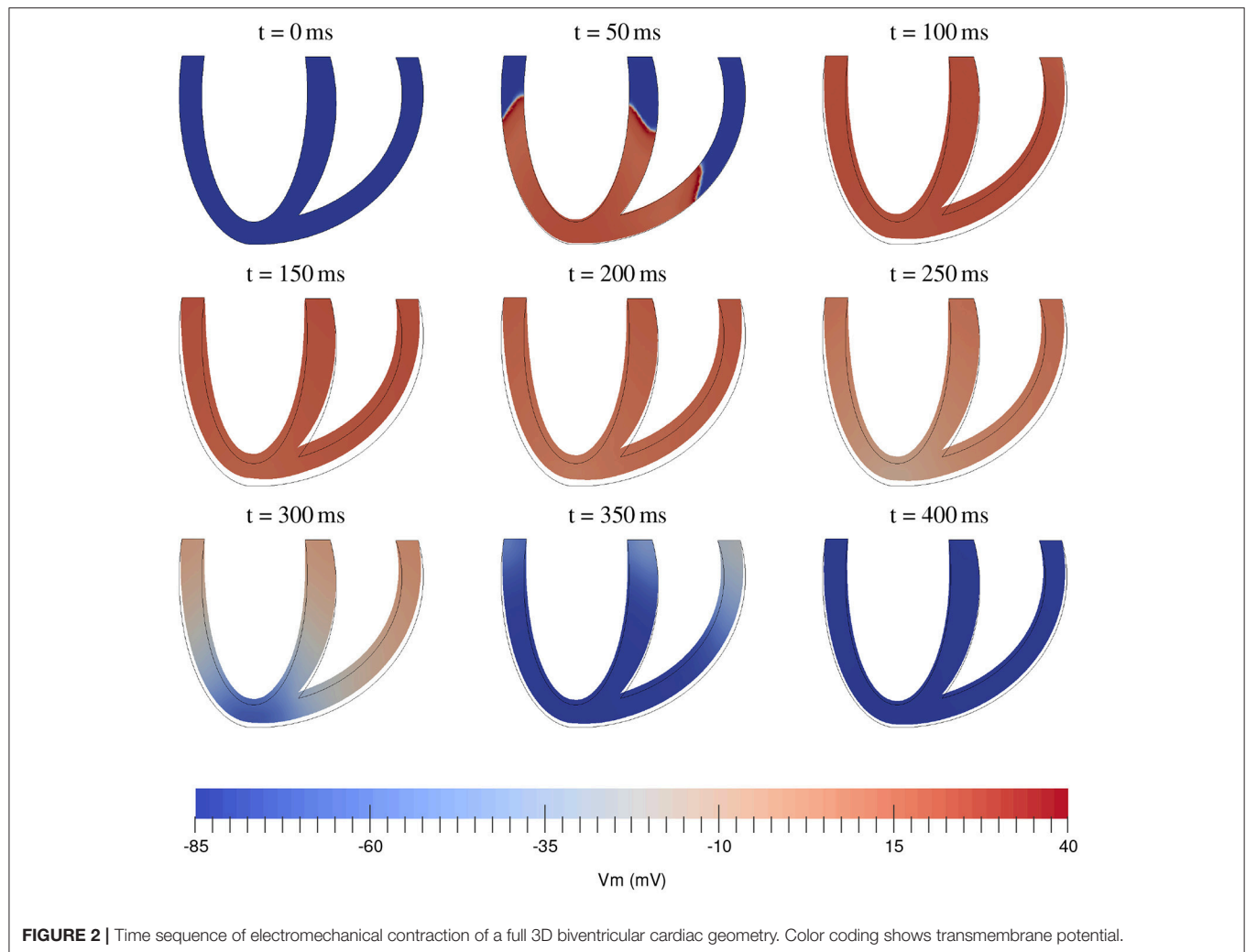


FIGURE 2 | Time sequence of electromechanical contraction of a full 3D biventricular cardiac geometry. Color coding shows transmembrane potential.

speed-up factor is the following: currently all field values and gradients at the quadrature points are calculated for each residual or Jacobian belonging to some field(s), for maximum flexibility. Thus, one may avoid unnecessary interpolation in order to accelerate the computation of the residuals: if the residual of field A is independent of field B, the value or gradient of field B at the quadrature points is not needed.

The use of PETSc enables to parallelize the computation on high-performance computing clusters (HPC). Smaller (test) runs, can be run on a desktop computer, requiring about 32GB of RAM memory to run the biventricular model in **Figure 2** with the TP06 cell model. There is no significant difference in memory cost between mono- and bidomain equations, since the latter introduces only few new state variables (extracellular potential, extracellular conductivities).

In this paper we have chosen to illustrate our approach using simple standard problems: the benchmark for electrophysiology (Niederer et al., 2011) and simple illustration of electromechanics for the fully coupled equations an idealized biventricular geometry. This is because we mainly wanted to describe of the methodology and place it to the existing environment and did

not focus on specific scientific applications. Such simulations can definitely be performed using our methodology and will be presented in subsequent papers.

AUTHOR CONTRIBUTIONS

AP and HD designed the research. SA implemented the methods. SA, HD, and AP wrote the manuscript.

FUNDING

SA was funded by BOF-Ghent. HD was funded by FWO-Flanders during part of this work.

ACKNOWLEDGMENTS

The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO and the Flemish Government—department EWI.

REFERENCES

- Abhyankar, S., Brown, J., Constantinescu, E., Ghosh, D., and Smith, B. F. (2014). *PETSc/TS: A Modern Scalable DAE/ODE Solver Library*. Preprint ANL/MCS-P5061-0114, ANL. Submitted to TOMS
- Adams, M. (2002). Evaluation of three unstructured multigrid methods on 3d finite element problems in solid mechanics. *Int. J. Numer. Methods Eng.* 55, 519–534. doi: 10.1002/nme.506
- Akenside, M. (1763). An account of a blow upon the heart, and of its effects: by mark akenside, mdfrs and physician to her majesty. *Philos. Trans.* 53, 353–355.
- Ascher, U. M., Ruuth, S. J., and Spiteri, R. J. (1997). Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Appl. Numer. Math.* 25, 151–167.
- Augustin, C. M., Neic, A., Liebmann, M., Prassl, A. J., Niederer, S. A., Haase, G., et al. (2016). Anatomically accurate high resolution modeling of human whole heart electromechanics: a strongly scalable algebraic multigrid solver method for nonlinear deformation. *J. Comput. Phys.* 305, 622–646. doi: 10.1016/j.jcp.2015.10.045
- Ayachit, U. (2015). *The Paraview Guide: A Parallel Visualization Application*. Clifton Park, NY: Kitware Inc.
- Bakir, A. A., and Dokos, S. (2015). “A gap junction-based cardiac electromechanics model,” in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE (Milano: IEEE)*, 25–28.
- Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., et al. (2016a). *PETSc Users Manual*. Technical Report ANL-95/11 - Revision 3.7, Argonne National Laboratory.
- Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., et al. (2016b). *PETSc Web Page*. Available online at: <http://www.mcs.anl.gov/petsc>
- Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F. (1997). “Efficient management of parallelism in object oriented numerical software libraries,” in *Modern Software Tools in Scientific Computing*, eds E. Arge, A. M. Bruaset, and H. P. Langtangen (Lemont, IL: Birkhäuser Press), 163–202.
- Barral, N., Knepley, M. G., Lange, M., Piggott, M. D., and Gorman, G. J. (2016). “Anisotropic mesh adaptation in Firedrake with PETSc DMplex,” in *25th International Meshing Roundtable*, eds S. Owen and H. Si (Washington, DC).
- Bayer, J., Blake, R., Plank, G., and Trayanova, N. (2012). A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models. *Ann. Biomed. Eng.* 40, 2243–2254. doi: 10.1007/s10439-012-0593-5
- Bernabeu, M. O., Pathmanathan, P., Pitt-Francis, J., and Kay, D. (2010). Stimulus protocol determines the most computationally efficient preconditioner for the bidomain equations. *IEEE Trans. Biomed. Eng.* 57, 2806–2815. doi: 10.1109/TBME.2010.2078817
- Boscarino, S., Pareschi, L., and Russo, G. (2013). Implicit-explicit runge-kutta schemes for hyperbolic systems and kinetic equations in the diffusion limit. *SIAM J. Sci. Comput.* 35, A22–A51. doi: 10.1137/110842855
- Brenner, S., and Scott, R. (2007). *The Mathematical Theory of Finite Element Methods*, Vol. 15. New York, NY: Springer Science & Business Media.
- Briggs, W. L., Henson, V. E., and McCormick, S. F. (2000). *A Multigrid Tutorial*. Philadelphia, PA: SIAM.
- Brown, J. (2010). Efficient nonlinear solvers for nodal high-order finite elements in 3d. *J. Sci. Comput.* 45, 48–63. doi: 10.1007/s10915-010-9396-8
- Brown, J., Knepley, M. G., May, D. A., McInnes, L. C., and Smith, B. (2012). “Composable linear solvers for multiphysics,” in *Parallel and Distributed Computing (ISPD), 2012 11th International Symposium on (Munich: IEEE)*, 55–62.
- Butcher, J. C. (2016). *Numerical Methods for Ordinary Differential Equations*. Hoboken, NJ: John Wiley & Sons.
- Childs, H., Brugger, E., Whitlock, B., Meredith, J., Ahern, S., Pugmire, D., et al. (2012). “VisIt: an end-user tool for visualizing and analyzing very large data,” in *High Performance Visualization—Enabling Extreme-Scale Scientific Insight* (New York, NY), 357–372.
- Ciarlet, P. G. (2002). *The Finite Element Method for Elliptic Problems*. Philadelphia, PA: SIAM.
- Colli Franzone, P., Pavarino, L. F., and Scacchi, S. (2016). Bioelectrical effects of mechanical feedbacks in a strongly coupled cardiac electro-mechanical model. *Math. Models Methods Appl. Sci.* 26, 27–57. doi: 10.1142/S0218202516500020
- Continuity (2018). *Continuity: A Problem Solving Environment for Multi-Scale Biology*. Available online at: <https://continuity.ucsd.edu/>
- Courant, R., Friedrichs, K., and Lewy, H. (1928). Über die partiellen differenzengleichungen der mathematischen physik. *Math. Annal.* 100, 32–74.
- FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophys. J.* 1, 445–466.
- Fluent, I. (2006). *Fluent 6.3 User's Guide*. Canonsburg, PA: Fluent documentation.
- Frankel, T. (2012). *The Geometry of Physics: An Introduction, 3rd Edn*. Cambridge; New York, NY: Cambridge University Press.
- Franzone, P. C., Pavarino, L. F., and Scacchi, S. (2015). Parallel multilevel solvers for the cardiac electro-mechanical coupling. *Appl. Numer. Math.* 95, 140–153. doi: 10.1016/j.apnum.2014.11.002
- Fritz, T., Wieners, C., Seemann, G., Steen, H., and Dössel, O. (2014). Simulation of the contraction of the ventricles in a human heart model including atria and pericardium. *Biomech. Model. Mechanobiol.* 13, 627. doi: 10.1007/s10237-013-0523-y
- Geuzaine, C., and Remacle, J.-F. (2009). Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Methods Eng.* 79, 1309–1331. doi: 10.1002/nme.2579
- Giraldo, F. X., Kelly, J. F., and Constantinescu, E. M. (2013). Implicit-explicit formulations of a three-dimensional nonhydrostatic unified model of the atmosphere (numa). *SIAM J. Sci. Comput.* 35, B1162–B1194. doi: 10.1137/120876034
- Göktepe, S., and Kuhl, E. (2010). Electromechanics of the heart: a unified approach to the strongly coupled excitation-contraction problem. *Comput. Mech.* 45, 227–243. doi: 10.1007/s00466-009-0434-z
- Göktepe, S., and Kuhl, E. (2010). Electromechanics of the heart: a unified approach to the strongly coupled excitation-contraction problem. *Computational Mechanics*, 45(2-3):227–243.
- Göktepe, S., Menzel, A., and Kuhl, E. (2013). “Micro-structurally based kinematic approaches to electromechanics of the heart. in *Computer Models in Biomechanics*, eds G. A. Holzapfel and E. Kuhl (Dordrecht: Springer), 175–187.
- Guccione, J. M., Costa, K. D., and McCulloch, A. D. (1995). Finite element stress analysis of left ventricular mechanics in the beating dog heart. *J. Biomech.* 28, 1167–1177.
- Gurev, V., Pathmanathan, P., Fattebert, J.-L., Wen, H.-F., Magerlein, J., Gray, R. A., et al. (2015). A high-resolution computational model of the deforming human heart. *Biomech. Model. Mechanobiol.* 14, 829–849. doi: 10.1007/s10237-014-0639-8
- Holzapfel, G. A., and Ogden, R. W. (2009). Constitutive modelling of passive myocardium: a structurally based framework for material characterization. *Philos. Trans. R. Soc. Lond. A Math. Phys. Eng. Sci.* 367, 3445–3475. doi: 10.1098/rsta.2009.0091
- Isaac, T., and Knepley, M. G. (2015). Support for non-conformal meshes in PETSc's DMplex interface. *arXiv [Preprint]*. arXiv:1508.02470.
- Itskov, M. (2013). *Tensor Algebra and Tensor Analysis for Engineers: With Applications to Continuum Mechanics, 3rd Edn*. Mathematical engineering. Heidelberg: Springer.
- Karniadakis, G., and Sherwin, S. (2013). *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford: Oxford University Press.
- Keener, J. P., and Sneyd, J. (2009). *Mathematical Physiology, Vol. 1*. New York, NY: Springer.
- Keldermann, R. H., Nash, M. P., Gelderblom, H., Wang, V. Y., and Panfilov, A. V. (2010). Electromechanical wavebreak in a model of the human left ventricle. *Am. J. Physiol. Heart Circ. Physiol.* 299, H134–H143. doi: 10.1152/ajpheart.00862.2009
- Kennedy, C., and Carpenter, M. (2001). Additive runge-kutta schemes for convection-diffusion-reaction equations. *Appl. Numer. Math.* 44, 139–181. doi: 10.1016/S0168-9274(02)00138-1
- Keyes, D. E., McInnes, L. C., Woodward, C., Gropp, W., Myra, E., Pernice, M., et al. (2013). Multiphysics simulations: Challenges and opportunities. *Int. J. High Perform. Comput. Appl.* 27, 4–83. doi: 10.1177/1094342012468181
- Kitware (2017). *The Extensible Data Model and Format*. Available online at: http://www.xdmf.org/index.php/XDMF_Model_and_Format
- Knepley, M. G., Brown, J., Rupp, K., and Smith, B. F. (2013). Achieving high performance with unified residual evaluation. *arXiv [Preprint]*. arXiv:1309.1204.

- Knepley, M. G., Lange, M., and Gorman, G. J. (2015). Unstructured overlapping mesh distribution in parallel. *arXiv [Preprint]*. arXiv:1506.06194.
- Kohl, P., Nesbitt, A. D., Cooper, P. J., and Lei, M. (2001). Sudden cardiac death by commotio cordis: role of mechano-electric feedback. *Cardiovasc. Res.* 50, 280–289. doi: 10.1016/S0008-6363(01)00194-8
- Lafortune, P., Aris, R., Vázquez, M., and Houzeaux, G. (2012). Coupled electromechanical model of the heart: parallel finite element formulation. *Int. J. Numer. Methods Biomed. Eng.* 28, 72–86. doi: 10.1002/cnm.1494
- Land, S., Niederer, S. A., and Smith, N. P. (2012). Efficient computational methods for strongly coupled cardiac electromechanics. *IEEE Trans. Biomed. Eng.* 59, 1219–1228. doi: 10.1109/TBME.2011.2112359
- Lange, M., Knepley, M. G., and Gorman, G. J. (2015). “Flexible, scalable mesh and data management using PETSc DMPlex,” in *Proceedings of the Exascale Applications and Software Conference* (Edinburgh).
- Liu, L. and Keyes, D. E. (2015). Field-split preconditioned inexact newton algorithms. *SIAM J. Sci. Comput.* 37, A1388–A1409. doi: 10.1137/140970379
- Maron, B. J. (2003). Sudden death in young athletes. *N. Engl. J. Med.* 349, 1064–1075. doi: 10.1056/NEJMra022783
- Maron, B. J. and Estes, N. A. M. III. (2010). Commotio cordis. *N. Engl. J. Med.* 362, 917–927. doi: 10.1056/NEJMra0910111
- Marsden, J. E., and Hughes, T. J. R. (1994). *Mathematical Foundations of Elasticity*. New York, NY: Dover.
- Meola, F. (1879). La commozione toracica. *Gior Internaz Sci. Med.* 1, 923–937.
- Mirams, G. R., Arthurs, C. J., Bernabeu, M. O., Bordas, R., Cooper, J., Corrias, A., et al. (2013). Chaste: an open source c++ library for computational physiology and biology. *PLoS Comput. Biol.* 9:e1002970 doi: 10.1371/journal.pcbi.1002970
- Nagumo, J., Arimoto, S., and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proc. IRE* 50, 2061–2070.
- Nash, M. P., and Hunter, P. J. (2000). Computational mechanics of the heart. *J. Elast. Phys. Sci. Solids* 61, 113–141. doi: 10.1023/A:1011084330767
- Nesbitt, A. D., Cooper, P. J., and Kohl, P. (2001). Rediscovering commotio cordis. *Lancet* 357:1195. doi: 10.1016/S0140-6736(00)04338-5
- Niederer, S., Hunter, P., and Smith, N. (2006). A quantitative analysis of cardiac myocyte relaxation: a simulation study. *Biophys. J.* 90, 1697–1722. doi: 10.1529/biophysj.105.069534
- Niederer, S. A., Kerfoot, E., Benson, A. P., Bernabeu, M. O., Bernus, O., Bradley, C., et al. (2011). Verification of cardiac tissue electrophysiology simulators using an n-version benchmark. *Philos. Trans. R. Soc. A* 369, 4331–4351. doi: 10.1098/rsta.2011.0139
- Niederer, S. A., and Smith, N. P. (2008). An improved numerical method for strong coupling of excitation and contraction models in the heart. *Prog. Biophys. Mol. Biol.* 96, 90–111. doi: 10.1016/j.pbiomolbio.2007.08.001
- Open CASCADE (2017). *MED Data Format*. Guyancourt.
- Ozakin, A., and Yavari, A. (2010). A geometric theory of thermal stresses. *J. Math. Phys.* 51:032902. doi: 10.1063/1.3313537
- Panfilov, A., Keldermann, R., and Nash, M. (2007). Drift and breakup of spiral waves in reaction–diffusion–mechanics systems. *Proc. Natl. Acad. Sci. U.S.A.* 104, 7922–7926. doi: 10.1073/pnas.0701895104
- Pathmanathan, P., and Whiteley, J. P. (2009). A numerical method for cardiac mechanoelectric simulations. *Ann. Biomed. Eng.* 37, 860–873. doi: 10.1007/s10439-009-9663-8
- Pavarino, L. F., and Scacchi, S. (2011). Parallel multilevel schwarz and block preconditioners for the bidomain parabolic-parabolic and parabolic-elliptic formulations. *SIAM J. Sci. Comput.* 33, 1897–1919. doi: 10.1137/100808721
- Pennacchio, M., and Simoncini, V. (2009). Algebraic multigrid preconditioners for the bidomain reaction–diffusion system. *Appl. Numer. Math.* 59, 3033–3050. doi: 10.1016/j.apnum.2009.08.001
- Plank, G., Prassl, A. J., and Augustin, C. (2013). Computational challenges in building multi-scale and multi-physics models of cardiac electro-mechanics. *Biomed. Eng.* doi: 10.1515/bmt-2013-4318. [Epub ahead of print].
- Plank, G., Zhou, L., Greenstein, J. L., Cortassa, S., Winslow, R. L., O’Rourke, B., et al. (2008). From mitochondrial ion channels to arrhythmias in the heart: computational techniques to bridge the spatio-temporal scales. *Philos. Trans. R. Soc. Lond. A Math. Phys. Eng. Sci.* 366, 3381–3409. doi: 10.1098/rsta.2008.0112
- Poirier, D., Allmaras, S. R., McCarthy, D. R., Smith, M. F., and Enomoto, F. Y. (1998). *The cgns System*. Albuquerque, NM: AIAA. Paper 98-3007.
- Quarteroni, A., and Valli, A. (1999). *Domain Decomposition Methods for Partial Differential Equations*. Number CMCS-BOOK-2009-019. Oxford: Oxford University Press.
- Quinn, T. A., Kohl, P., and Ravens, U. (2014). Cardiac mechano-electric coupling research: fifty years of progress and scientific innovation. *Prog. Biophys. Mol. Biol.* 115, 71–75. doi: 10.1016/j.pbiomolbio.2014.06.007
- Rokos, G. and Gorman, G. (2013). “Pragmatic - parallel anisotropic adaptive mesh toolkit,” in *Facing the Multicore-Challenge III*, vol. 7686; *Lecture Notes in Computer Science*, eds R. Keller, D. Kramer, and J.-P. Weiss (Berlin; Heidelberg: Springer), 143–144.
- Rossi, S., Lassila, T., Ruiz-Baier, R., Sequeira, A., and Quarteroni, A. (2014). Thermodynamically consistent orthotropic activation model capturing ventricular systolic wall thickening in cardiac electromechanics. *Eur. J. Mech. A Solids* 48, 129–142. doi: 10.1016/j.euromechsol.2013.10.009
- Sachse, F. B. (2004). *Computational Cardiology: Modeling of Anatomy, Electrophysiology, and Mechanics*. Number 2966 in LNCS, Tutorial. Berlin; New York, NY: Springer.
- Schoof, L. A., and Yarberr, V. R. (1994). *EXODUS II: A Finite Element Data Model*. Technical Report SAND92-2137, Sandia National Laboratories, Albuquerque, NM.
- Smith, B., Björstad, P., and Gropp, W. (2004). *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge: Cambridge University Press.
- Smith, G. F. (2012). *Constitutive Equations for Anisotropic and Isotropic Materials*, volume 3. Newnes.
- Söderlind, G., Jay, L., and Calvo, M. (2015). Stiffness 1952–2012: Sixty years in search of a definition. *BIT Numer. Math.* 55, 531–558. doi: 10.1007/s10543-014-0503-3
- Spiteri, R. J., and Dean, R. C. (2008). On the performance of an implicit–explicit runge–kutta method in models of cardiac electrical activity. *IEEE Trans. Biomed. Eng.* 55, 1488–1495. doi: 10.1109/TBME.2007.914677
- Sundnes, J., Lines, G., Mardal, K., and Tveito, A. (2002). Multigrid block preconditioning for a coupled system of partial differential equations modeling the electrical activity in the heart. *Comput. Methods Biomech. Biomed. Eng.* 5, 397–409. doi: 10.1080/1025584021000025023
- ten Tusscher, K. H., and Panfilov, A. V. (2006). Alternans and spiral breakup in a human ventricular tissue model. *Am. J. Physiol. Heart Circ. Physiol.* 291, H1088–H1100. doi: 10.1152/ajpheart.00109.2006
- The HDF Group (1997–2017). *Hierarchical Data Format, version 5*. Champaign, IL.
- Trottenberg, U., Oosterlee, C. W., and Schuller, A. (2000). *Multigrid*. Orlando, FL: Academic Press.
- Van, P., Brezina, M., Mandel, J., et al. (2001). Convergence of algebraic multigrid based on smoothed aggregation. *Numer. Math.* 88, 559–579.
- Vaněk, P., Mandel, J., and Brezina, M. (1996). Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing* 56, 179–196.
- Whiteley, J. P., Bishop, M. J., and Gavaghan, D. J. (2007). Soft tissue modelling of cardiac fibres for use in coupled mechano-electric simulations. *Bull. Math. Biol.* 69, 2199–2225. doi: 10.1007/s11538-007-9213-1
- Wikipedia (2017). *PLY (File Format)* — Wikipedia. The Free Encyclopedia.
- Yavari, A. (2010). A geometric theory of growth mechanics. *J. Nonlinear Sci.* 20, 781–830. doi: 10.1007/s00332-010-9073-y
- Zienkiewicz, O. C., Taylor, R. L., and Zhu, J. Z. (2013). *The Finite Element Method: Its Basis and Fundamentals, 7th Edn*. Amsterdam: Elsevier; Butterworth-Heinemann.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2018 Arens, Dierckx and Panfilov. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

NOTATION

A	acceleration
B	body force
\mathcal{B}	material manifold
C	deformation tensor
E	strain tensor
$\{E_A\}_{A \in \{F, S, N\}}$	material fiber directions
F	deformation gradient
G	material metric
g	spatial metric
J	Jacobian of deformation
P	first Piola-Kirchhoff stress tensor
P_{appl}	applied pressure
R	reaction rates for internal variables
S	second Piola-Kirchhoff stress tensor
\mathcal{S}	spatial manifold
Σ_e	extracellular conduction tensor
Σ_i	intracellular conduction tensor
V	velocity
V_e	extracellular voltage
V_m	transmembrane voltage
X	material coordinates
x	spatial coordinates
Γ	internal variables (i.e., ionic concentrations, gating variables, tensions variables)
ϕ	deformation field